

# 유전자 알고리즘을 이용한 블럭단위의 설비배치에 관한 연구

우성식, 박양병  
경희대학교 산업공학과

## <Abstract>

The most research on facility layout problems ignored the actual shape of building where the activities(departments) are to be arranged. They also ignored the aisles between departments inside the building.

In this paper, we present a genetic algorithm that searches a very good facility layout with horizontal aisles for two different cases with respect to the department shape.

From the extensive experiments, the proposed genetic algorithm generated better layouts than the ones obtained by applying Tam's algorithm. It showed about 10% improvement of performance. We found out the best combination of genetic operators through the experiments.

## 1. 서론

설비배치 문제는 물자의 흐름량과 각 활동(부서)간 거리 곱의 총합을 최소화하는 활동들의 배열을 결정하는 문제이다[6]. 각 활동간 물자의 흐름량은 고정되어 있고 활동간 거리는 배치의 순서에 따라 다르므로 물자의 흐름량이 큰 활동공간들은 가깝게 위치시켜야 한다. 설비배치 문제는 수식에서 보듯이 두 활동간의 가중치가 클수록 기본적으로 두 활동공간은 근접시켜야 한다. 설비배치 문제는 활동공간의 수가 증가할수록 계산량이 지수적으로 증가하는 NP-complete 문제에 속한다.

물자의 흐름량을 가중치라 정의하고 정량적인 면만을 고려한 배치문제의 목적함수는 아래 식과 같다.

$$\text{Min. } F(x) = \sum_{i=1}^n \sum_{j \neq i}^n w_{ij} \cdot d_{ij} \quad (1)$$

여기서

$F(x)$  : 총 이동거리

$n$  : 활동공간(부서)의 개수

$w_{ij}$  : 활동공간  $i$  와  $j$  사이의 흐름량

$d_{ij}$  : 활동공간  $i$  와  $j$  사이의 직각거리

본 연구에서는 유전자 알고리즘을 적용하여 최적의 배치를 탐색하는 새로운 방법을 소개한다. 그리고 여러 가지 유전자 연산자를 적용하여 비교하였다[2].

배치문제에 대한 기존의 연구에서는 모든 활동공간의 형태를 동일한 크기의 정사각형으로 가정하거나[6] 또는 고정된 면적에 대해 가로와 세로 비율의 하한치와 상한치를 설정하여 배치문제를 해결하려 하였다[5]. 또한 공간의 효율성만을 고려한 기존의 연구에서는 대부분 배치공간의 실제 형태와 통로를 배치과정에서 무시하였다.

본 연구에서는 이러한 문제를 해결하기 위해 설계자가 배치할 전체공간의 가로와 세로의 크기를 정해 주면 블럭단위로 가로(수평)방향의 통로가 자동적으로 생성되면서 각 활동공간을 효율적으로 배치하는 유전자 알고리즘을 개발하였다.

## 2. 블럭단위의 설비배치

활동공간을 배치함에 있어서 구축과정에서 가로방향의 통로가 생성되도록 하였다. 각 활동공간은 직사각형으로 가정하였으며 거리는 중심점 사이의 직각거리로 가정하였다. 각 활동간의 흐름량은 사전에 주어진다. 설계자가 배치할 전체영역의 우측 상한치를 정해 주면 순서대로 활동공간을 배치하다가 우측 상한치를 초과할 경우 통로가 생성되면서 새로운 열

의 좌측부터 다시 배치해간다. 여기서 하나의 통로가 생성되면서 배치되는 공간을 블럭이라 하였다.

각 활동공간의 크기는 가로와 세로가 고정되어 있는 경우와 면적이 주어지고 가로/세로 비율의 상한치가 주어져 있는 경우로 나누었다. 이러한 두 가지 경우에 대해 제안된 유전자 알고리즘을 각각 적용하여 배치를 형성하였다. 두 경우에 배치를 형성하는 과정(배치방법)을 요약하면 다음과 같다.

#### (i) 각 활동공간의 가로와 세로의 크기가 고정되어 있는 경우

순차적으로 배치할 때 블럭의 폭(세로)은 인접 활동공간 중 세로의 길이가 가장 큰 것을 기준으로 다음 통로가 생성되도록 하였다. 그러므로 블럭의 폭은 일정하지 않다.

#### (ii) 각 활동공간의 가로/세로 비율의 상한치가 주어지는 경우

이 경우에는 설계자가 미리 블럭의 폭을 정해 주면 각 활동공간의 세로는 블럭의 폭이 되고 가로는 주어진 면적에 의해 계산된다. 계산된 가로의 길이가 가로/세로 비율의 상한치를 초과하면 가로/세로 비율의 상한치에 맞게 가로와 세로가 계산되어 배치된다. 따라서 이 경우에는 블럭의 폭은 일정하게 유지된다.

### 3. 유전자 알고리즘의 적용

유전자 알고리즘은 적자생존의 원리를 이용하여 탐색기법에 적용한 것으로 보다 더 나은 해를 찾는 데 매우 뛰어난 성능을 보여왔다 [1]. 유전자 알고리즘은 해의 표현과 각 연산자의 적용에 따라 그 성능이 다르게 나타난다.

#### (1) 해(string)의 표현

배치의 해는 string으로 표현하며, string내의 번호는 활동공간들의 배치순서에 따라 표기된다.

#### (2) 교차변이(crossover) 연산자

해의 활동공간 번호가 중복되지 않도록 하는 변형된 교차변이 연산자가 필요하다. 이중 PMX, OX, CX 기법[1]들을 적용하였다.

#### (3) 돌연변이(mutation) 연산자

유전자 알고리즘을 적용할 때 빠른 수렴(premature)을 막기 위한 보험정책으로 돌연변이 연산자를 적용한다. 돌연변이 연산자는 inversion, insertion, swap 기법[7]들을 적용하였다.

#### (4) 재생규칙(reproduction rule)

유전자 알고리즘을 적용할 때 다음 세대에 더 좋은 유전자가 복사되도록 하기 위해서 재생이 필요하다. 재생규칙 중에서 selection, ranking, tournament 기법[7]들을 적용했다.

① **selection** : 설비배치 문제는 최소화 문제이므로 목적함수값이 작은 string을 다음 세대에 복사한다. 각 string의 목적함수값과 각 목적함수들의 평균값의 비율로 다음 세대에 복사한다. 목적함수는 식(1)을 사용한다.

② **ranking** : 각 string의 목적함수값의 순위를 계산하여 순위에 따라 결정된 갯수를 다음 세대에 복사한다. 이때 선형식의 ranking 방법을 사용하였으며 scaling 파라미터는  $q = 0.15$ ,  $r = 0.01$ 로 설정하였다.

③ **tournament** : 각 string의 목적함수값을 2개씩 비교하여 이중 목적함수값이 작은 것을 다음 세대에 2개 복사한다.

#### (5) 적합도 평가(fitness evaluation)

각 세대에서 가장 작은 목적함수값을 정해진 세대까지 비교하여 구한다.

#### (6) 유전연산과 발생확률

실험에서 교차변이 확률과 돌연변이의 확률은 0.1~1.0 사이의 값을 산정하였고 각 operator들을 조합하여 수행도를 측정하여 가장 뛰어난 성능의 조합과 확률을 찾았다.

#### (7) 초기해와 해집단(population)

초기해는 2개의 string을 임의로 다르게 10개씩 생성하였다.

#### (8) 종료조건

5000 generations에서 실험을 종료하도록 하였다.

### 4. 계산실험

제기된 유전자 알고리즘은 C++ 언어로 프로그래밍하였으며 펜티엄급 586 PC를 이용하여

활동공간 갯수	배치 방법	최적해	제안된 유전자 알고리즘에 의한 해	최적성
5	(i)	512	512*	100%
	(ii)	533	533*	100%
6	(i)	896	896*	100%
	(ii)	884	884*	100%
7	(i)	1319	1319*	100%
	(ii)	1230	1230*	100%
8	(i)	1940	1965	99.98%
	(ii)	1806	1806*	100%
9	(i)	2673	2673*	100%
	(ii)	2664	2644*	100%
10	(i)	3709	3709*	100%
	(ii)	3519	3591	99.98%

<표 1> 활동공간이 5~10개인 문제에서 제안된 알고리즘에 의한 해와 최적해의 총 이동거리 비교

활동 공간 갯수	배치 방법	제안된 GA에 의한 해	Tam의 알고리즘에 의한 해	GA수행도 효율
15	(i)	9120	13762	33.7% ↓
	(ii)	9855	12240	19.4% ↓
20	(i)	21885	26921	18.7% ↓
	(ii)	22656	28646	20.9% ↓
30	(i)	50492	55668	9.3% ↓
	(ii)	52885	58824	10% ↓

<표 2> 활동공간이 15~30개인 문제에서 제안된 알고리즘에 의한 해와 Tam의 해의 총 이동거리 비교

배치 방법	tournament			
		inversion	insertion	swap
(i)	PMX	56067	50492*	51242
	OX	55494	53963	51294
	CX	54129	55017	51967
(ii)	PMX	55861	54989	53125
	OX	56119	53096	54187
	CX	56130	53194	52869*

<표 3> 유전연산자들의 수행도(총 이동거리) 비교

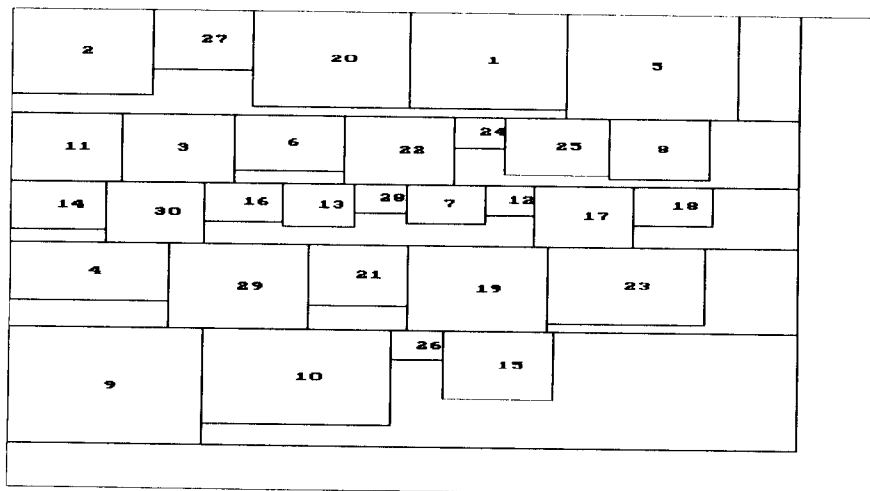
실험하였다. 활동공간의 갯수가 5~30개인 문제들을 만들어 실험하였다.

먼저 PC의 성능상, 활동공간의 모든 조합에 대한 전수검사가 가능한 5~10개 활동공간 문제의 최적값과 개발된 알고리즘의 해를 비교하였다. 이때 우측 상한치는 거리 35로 설정하였다. 그 결과는 <표 1>에 나타나있다.

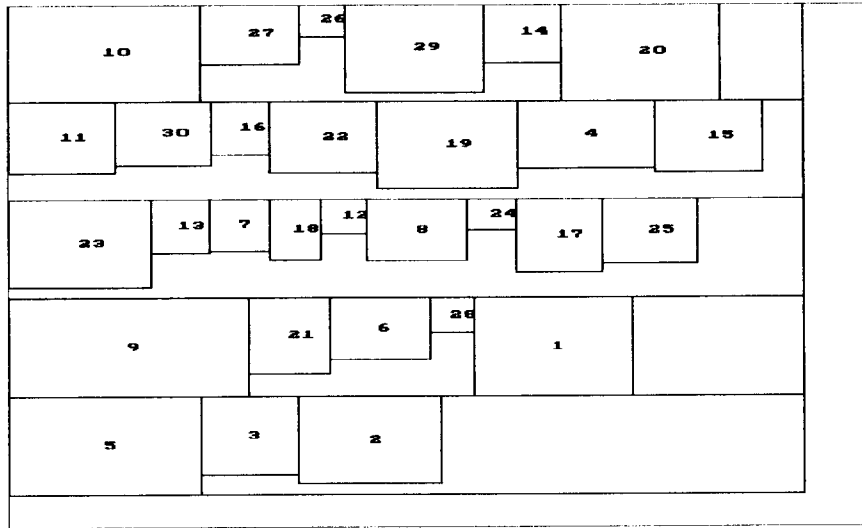
PC의 성능상 전수검사가 어려운 15~30개 활동공간의 문제들에 대한 비교는 <표 2>에서와 같이 Tam의 알고리즘의 결과와 비교하였다.

흐름량의 자료는 Nugent et al.[4]의 논문에 정리되어 있는 자료를 사용하였다. (i)과 (ii)에 사용된 활동공간의 크기는 Tam[5]의 논문에 소개된 자료를 사용하였다.

유전자 알고리즘의 적용을 위한 교차변이, 돌연변이 연산자와 재생규칙의 결정은 실험을 통해서 이루어졌다. 계산실험은 (i)과 (ii)의 두 가지 경우로 나누어서 실험하였다. <표 3>에



<그림 1> 30개 활동공간의 가로와 세로의 크기가 주어진 경우의 배치 결과



<그림 2> 30개 활동공간의 가로/세로 비율이 주어진 경우의 배치 결과

서와 같이 30개의 활동공간을 실험할 때 (i)의 배치는 교체변이 연산자는 확률 0.25일 때 PMX, 돌연변이 연산자는 확률 0.1일 때 insertion, 재생규칙은 tournament가 가장 좋은 수행도를 보였다. (ii)의 배치는 블럭의 폭을 10으로 고정하였으며 교체변이 연산자는 확률 0.1일 때 OX, 돌연변이 연산자는 확률 0.1일 때 swap, 재생규칙은 tournament가 가장 좋은 수행도를 나타냈다. <그림 1, 2>는 (i)(ii)의 배치에 대해 30개 활동공간을 가정했을 때 최적의 배치결과를 보여준다.

## 5. 결론

본 논문에서는 배치설계자가 배치할 영역의 우측 상한치를 설정해주면 순차적으로 수평방향의 통로와 함께 활동공간들을 배치해 주는 새로운 배치방법과 최적의 배치대안을 탐색해주는 유전자 알고리즘을 제안한다.

실험결과, 최적의 배치를 형성시키는 연산자들의 조합과 확률을 찾아내었다. 제안된 알고리즘은 최적해를 탐색하는데 뛰어난 성능을 보였다. 또한, Tam의 알고리즘에 의한 해와 비교할 때 약 10% 이상의 수행도 개선효과를 보였다.

본 연구에서는 수평방향의 통로만이 생성되도록 하였는데 추후의 연구에서는 수평과 수직방향의 통로를 모두 고려하는 알고리즘의 연구가 필요할 것이다.

## < 참고문헌 >

- [1] D.E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison Wesley, 1989.
- [2] K.C.Chan, and H. Tansri, "A Study of Genetic Crossover operations on facilities layout problem", Computers industrial Engineering, Vol.26(1994), 557 - 550.
- [3] Kusiak, A., and Heragu. S. S, " The facility layout problem", European Journal of Operations Research, Vol. 29(1987), 229-251.
- [4] Nugent, C. E., Vollman, T. E, and Ruml. J., "An experimental comparison of techniques for the assignment of facilities to locations", Operations Research, Vol.16(1968), 150-173.
- [5] Tam. K. Y., "Genetic algorithms, function optimization, and facility layout design", European Journal of Operation Research, Vol.63(1992), 322-346.
- [6] White, J. A., Facilities Layout and Location : An Analytical approach., 2nd ed., Prentice Hall, NJ, 1992
- [7] Zbigniew Michalewicz, Genetic Algorithms +Data structure = Evolution Programs, 2nd ed., Springer-Verlag, 1994.